REMARKS

By this amendment, Claims 13-17 have been cancelled. Claims 1-12 have been amended to more clearly recite subject matter for which the Applicant desires patent protection. Claim 18 has been added. Hence, Claims 1-12 and 18 are pending in the application.

REQUEST TO CHANGE ATTORNEY DOCKET NUMBER

Applicant respectfully requests that the Attorney Docket Number be changed from "50277-456 [OID-1999-048-0" to "50277-0404."

SUMMARY OF THE REJECTIONS

Claims 1-17 have been rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over U.S. Patent Number 6,374,256 issued to Ng et al ("*Ng*") in view of U.S. Patent Number 6,321,261 issued to Glass ("*Glass*").

The rejections are respectfully traversed.

EACH OF THE PENDING CLAIMS IS PATENTABLE OVER THE CITED ART

Even if the cited references were to be properly combined, each of the pending claims features at least one element that is not disclosed, taught, or suggested by the cited art, either taken individually or in combination.

Claim 1

Independent Claim 1 features:

> A method for persistently storing an object belonging to a class, wherein the method comprises performing a machine-executed operation involving instructions, wherein the machine-executed operation is at least one of:
> a) sending said instructions over transmission media;
> b) receiving said instructions over transmission media;
> c) storing said instructions onto a machine-readable storage medium; and
> d) executing the instructions;
> wherein said instructions are instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

> **in response to receiving a request to instantiate a persistent agent for said class, performing the steps of:**
>
> (a) creating one or more structures within a persistent object store, wherein said one or more structures have portions that correspond to respective fields of objects belonging to said class; and
>
> (b) **instantiating a class-specific persistent agent for storing objects that belong to said class, wherein the class-specific persistent agent provides an interface, that includes a routine, for persistently storing the object in said structures within said persistent object store, wherein said class-specific persistent agent may only be used for operations involving objects belonging to said class**; and
>
> **in response to invoking the routine, the class-specific persistent agent storing the object in said one or more structures within the persistent object store, wherein the object is not derived from a persistent object base class.**

At least the above-bolded elements are not disclosed, taught, or suggested by the *Ng* or *Glass*, either individually or in combination.

Claim 1 describes an approach for persistently storing an object belonging to a class. According to Claim 1, in response to a receiving a request to instantiate a persistent agent for a class, one or more structures are created within a persistent object store. The one or more structures have portions that correspond to respective fields of objects belonging to the class. Further, a class-specific persistent agent for storing objects of the particular class is instantiated. The class-specific persistent agent provides an interface, which includes a routine, for persistently storing objects in the one or more structures within the persistent object store. The class-specific persistent agent may only be used for operations involving objects belonging to the class. In response to invoking the routine, the class-specific persistent agent stores the object in the one or more structures within the persistent object store. The object that is stored is not derived from a persistent object base class.

On the other hand, *Ng* describes an approach for creating indexes, in a relational database, that correspond to classes in an object-oriented application (title). *Ng* teaches that a mapping tool 508 maps each class of an object-oriented application to a table in a database (Abstract; Col. 7, lines 25-46). To store data in the relational database, the mapping tool uses a database connectively module (DBC) 208. The DBC 208 stores data in the relational database by issuing SQL statements to the database (Col. 6, lines 29-30).

Unfortunately, the mapping tool 508 of *Ng* may only be used with objects that are persistent-capable. For example, *Ng* states "a user provides tool 508 with persistence-capable-compatible classes in class library 506 for mapping to database 502" (Col. 7, lines 34-36). Thus, *Ng* suffers from the same deficiencies discussed in the Applicant's background on page 3, lines 14-24, namely that the programmer designing the classes for persistent objects must derive such classes from a persistent object base class.

Further, the mapping tool 508 and DBC 208 of *Ng* are used for storing all types of objects in the relational database. On the other hand, the class-specific specific agent of Claim 1 may only be used for operations involving objects belonging to a particular class.

In view of the fundamental differences discussed above, numerous elements of Claim 1 are not disclosed, taught, or suggested by *Ng* or *Glass*, either individually or in combination. For example, Claim 1 features the element of "in response to invoking the routine, the class-specific persistent agent storing the object in said one or more structures within the persistent object store, wherein the object is not derived from a persistent object base class." In sharp contrast to this element, *Ng* requires that the "user provide[s] tool 508 with persistence-capable-compatible classes in class library 506 for mapping to database 502" (See Col. 7, lines 34-36). Thus, the feature of "wherein the object is not derived from a persistent object base class" is not met by *Ng*. Further, no portion of *Glass* teaches or suggests this element. Consequently, even if *Glass* and *Ng* were to be properly combined, the resulting combination would fail to disclose, teach, or suggest this element.

Further, Claim 1 features "in response to receiving a request to instantiate a persistent agent for said class, performing the steps of:...(b) instantiating a class-specific persistent agent for storing objects that belong to said class, wherein the class-specific persistent agent provides an interface, that includes a routine, for persistently storing the object in said structures within said persistent object store, wherein said class-specific persistent agent may only be used for operations involving objects belonging to said class."

The Office Action appears to argue that either mapping tool 508 or DBC 208 is analogous to the claimed class-specific persistent agent. However, *Ng* teaches that the mapping tool 508 is used for all possible persistent-capable-compatible classes, and so the feature of "may only be used for operations involving objects belonging to said class" is not satisfied by the mapping tool 508.

Also, *Ng* lacks any suggestion of instantiating the mapping tool 508 in response to a receiving a request to do so. For example, FIG. 6 of *Ng* is a flowchart describing how to translate a class in an object-oriented application to a table entry in a database. The first step of FIG. 6 (step 602) assumes the existence of the tool 508, so the approach of *Ng* requires that the tool 508 be instantiated prior to first user interaction with the tool 508, namely the interaction of step 602. Thus, the feature of "in response to receiving a request to instantiate a persistent agent for said class" is not disclosed, taught, or suggested by *Ng*. While *Glass* discusses constructing a remote object (See Col. 3, lines 58-61), nothing in *Glass* suggests instantiating a class-specific persistent agent as claimed. Thus, even if *Ng* were to be properly combined with *Glass*, the resulting combination would fail to disclose, teach, or suggest this quoted claim element.

Similarly, the DBC 208 of *Ng* cannot be analogous to a class-specific persistent agent as claimed. The DBC 208 is used for all possible persistent-capable-compatible classes, and so the feature of "may only be used for operations involving objects belonging to said class" is not satisfied by the DBC 208.

Further, no portion of either *Ng* or *Glass* discuss instantiating the DBC 208 in response to receiving a user request to do so. To illustrate, the Office Action alleges that *Ng* teaches that "agents [can] be instantiated" by citing Col. 7., lines 9-10 of *Ng*; however, this portion lacks any suggestion of instantiating a class-specific persistent agent as claimed, but instead discusses generating persistence-capable objects that do not meet the express limitations of a class-specific persistent agent as claimed. For example, the persistence-capable objects discussed in this portion may be stored according to the approach of *Ng*, but such objects do not provide an interface, that includes a routine, for persistently storing the object in said structures within said persistent object store, wherein the class-specific persistent agent may only be used for operations involving objects belonging to a specific class. Thus, this element cannot be disclosed, taught, or suggested by *Ng*. Similarly, *Glass* fails to teach or suggest this element as well.

The Office Action states that it would have been obvious to "to have modified *Ng*'s invention, to include the details regarding the naming of created persistent agents, because *Ng* recognized the need to improve the techniques for mapping object-oriented application and database (Col. 2, lines 5-10), while *Glass* recognizes the need for supporting communications (Col. 1, lines 35-42)." However, notwithstanding the fact that neither *Ng* nor *Glass* discloses numerous claimed features, the Applicant respectfully submits that there is nothing in either *Ng* or *Glass* that teaches or suggests combining their respective teachings.

As stated in the Federal Circuit decision *In re Dembiczak*, 50 USPQ.2d 1617 (Fed. Cir. 1999), (citing *Gore v. Garlock*, 220 USPQ 303, 313 (Fed. Cir. 1983)), "it is very easy to fall victim to the insidious effect of the hindsight syndrome where that which only the inventor taught is used against its teacher." *Id.* The Federal Circuit stated in *Dembiczak* "that the best defense against subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or suggestion to combine prior art references." *Id.* Thus, the Federal Circuit explains that a proper obviousness analysis requires "***particular factual findings*** regarding the locus of the suggestion, teaching, or motivation to combine prior art references." *Id.* (emphasis added).

In particular, the Federal Circuit states:

> "We have noted that evidence of a suggestion, teaching, or motivation to combine may flow from the prior art references themselves, the knowledge of one of ordinary skill in the art, or, in some cases, from the nature of the problem to be solved...although 'the suggestion more often comes from the teachings of the pertinent references'...The range of sources available, however, does ***not diminish the requirement for actual evidence***. That is, the ***showing must be clear and particular***...Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence.'" *Id.* (emphasis added; internal citations omitted).

Neither *Ng* or *Glass* show any suggestion, teaching, or motivation to combine their teachings, nor does the Office Action provide a "clear and particular" showing of the suggestion, teaching, or motivation to combine their teachings. In fact, the portions of *Ng* and *Glass* cited by the Office Action to show a motivation for combination merely state:

> There are numerous challenges to improve the techniques for mapping object-oriented applications and databases in the object-database mapping tools. These techniques need to account for the organization of data in an object-oriented application compared with the organization of data in a database. (*Ng* at Col. 2, lines 5-10).
>
> With the rise of distributed systems, client/server computing, and internet/intranet interactions, inter-node communications between applications have become a prerequisite. Early operating systems lacked support for interapplication communications, forcing software developers to write custom code to perform remote procedure call (RPC) for each and every application that needed remote communications. (*Glass* at Col. 1, lines 35-42).

These cited portions do not suggest combining the teachings of either *Ng* or *Glass* with the other. At best, the cited portions discuss or lament a state of technology, but no motivation is provided in these cited portions as to why one would be motivated to combine *Ng* with *Glass* to produce the approach featured in the pending claims. It is respectfully submitted that such a hindsight observation is not consistent with the Federal Circuit's requirement for "particular factual findings."

As explained above, at least one element of Claim 1 is not disclosed, taught, or suggested by *Ng* or *Glass*, either individually or in combination. Further, *Ng* and *Glass* have not been properly combined. Consequently, it is respectfully submitted that Claim 1 is patentable over the cited art and is in condition for allowance.

Claim 11

Independent Claim 11 features:

> A method for retrieving a set of objects from a persistent object store, wherein the method comprises performing a machine-executed operation involving instructions, wherein the machine-executed operation is at least one of:
> a) sending said instructions over transmission media;
> b) receiving said instructions over transmission media;
> c) storing said instructions onto a machine-readable storage medium; and
> d) executing the instructions;
>
> wherein said instructions are instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

> **in response to a receiving a request to instantiate a persistent agent for a particular class, instantiating a class-specific persistent agent for retrieving objects of said particular class, said class-specific persistent agent provides an interface, that includes a routine, for retrieving objects of said particular class from the persistent object store, wherein the class-specific persistent agent may only be used for operations involving objects that belong to said particular class**; and
>
> in response to invoking the routine, the class-specific persistent agent retrieving the set of objects from the persistent object store.

At least the above-bolded elements are not disclosed, taught, or suggested by the *Ng* or *Glass*, either individually or in combination.

Claim 11 recites the feature of "in response to a receiving a request to instantiate a persistent agent for a particular class, instantiating a class-specific persistent agent for retrieving objects of said particular class, said class-specific persistent agent provides an interface, that includes a routine, for retrieving objects of said particular class from the persistent object store, wherein the class-specific persistent agent may only be used for operations involving objects that belong to said particular class." As explained above with reference to Claim 1, neither *Ng* nor *Glass* disclose, teach, or suggest a class-specific persistent may that may only be used for operations involving objects that belong to a particular class. As a result, this element is not disclosed, taught, or suggested by *Ng* nor *Glass*, taken individually or in combination.

At least one element of Claim 11 is not disclosed, taught, or suggested by *Ng* or *Glass*, either individually or in combination. Further, as explained above, *Ng* and *Glass* have not been properly combined. Consequently, it is respectfully submitted that Claim 11 is patentable over the cited art and is in condition for allowance.

Claims 2-10, 12, and 18

Claims 2-10, 12, and 18 are dependent claims, each of which depends (directly or indirectly) on one of the claims discussed above. Each of Claims 2-10, 12, and 18 is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of Claims 2-10, 12, and 18 introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those limitations is not included at this time, although the Applicants reserve the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

The fee for the two-month extension of time under 37 C.F.R. § 1.17(a) accompanies this response. To the extent necessary, a petition for any further extensions of time under 37 C.F.R. § 1.136 is hereby made. Please charge any necessary fees shortages or credit any overages to Deposit Account No. 50-1302.

Respectfully submitted,
HICKMAN PALERMO TRUONG & BECKER LLP

Christopher J. Brokaw
Reg. No. 45,620
Date: **September 6, 2005**

2055 Gateway Place Suite 500
San Jose, California 95110-1089
(408) 414-1225
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: **Mail Stop AF**, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

On September 6, 2005 By Angelica Maloney
Angelica Maloney